DETECTION AND INTERPRETATION OF TRANSPLANTED POSITIONS ON FULL-COLOR IMAGE FOR RICE PADDIES WITH FPGA

Yi-Syun Wang[1,a], Yu-Che Tsai[1,b], Yih-Shyh Chiou[*2,c], Mu-Jan Shih[3,d], You-Cheng Chen[4,e]

[1]Undergraduate Student, Department of Electronic Engineering, Chung Yuan Christian University
Taoyuan 320314, Taiwan
Email: [a]stanley900903@gmail.com, [b]jacky901219@gmail.com

[2]Associate Professor, Department of Electronic Engineering, Chung Yuan Christian University
Taoyuan 320314, Taiwan
Email: [c]choice@cycu.org.tw

[3]Graduate Student, Miin Wu School of Computing, National Cheng Kung University
Tainan 701401, Taiwan
Email: [d]shihmujan@gmail.com

[4]Graduate Student, Department of Electronic Engineering, Chung Yuan Christian University
Taoyuan 320314, Taiwan
Email: [e]blazecrasher621@gmail.com

KEY WORDS: Aerial Photographs, Convolutional Neural Network, Deep Learning, Field Programmable Gate Array

ABSTRACT:
In recent years, the satellite industry has undergone rapid development, with the emergence of "Starlink", which has played a significant role in driving this progress. Among various types of satellites, Low Earth Orbit (LEO) satellites have garnered substantial attention. These satellites are primarily employed for communication and remote sensing purposes. Remote sensing utilizes imaging techniques for environmental monitoring, weather observation, ocean current analysis, and more. With the continuous increase in the number of satellites, the applications of remote sensing technology have also expanded. This article employs deep learning and appropriate algorithms to investigate the developmental status of rice plants using full-color images captured by unmanned aerial vehicles (UAV). A Convolutional Neural Network (CNN) model is established to locate and label the positions of rice plants. Validation is conducted using an 80/20 training/testing data split ratio, and the developed model is optimized using Field-Programmable Gate Array (FPGA) technology. The efficiency is further verified using a known dataset of rice plant coordinates. The model achieves an accuracy of around 94% in terms of the F_1-measure value when using desktop computers. With the assistance of PYNQ-Z1, we successfully reduced power consumption to 1.66 watts while accepting a slight decrease in accuracy of approximately 1%. On the PYNQ-Z1, the computation speed increased by 120 times when utilizing Programmable Logic (PL) in addition to the Processing System (PS), compared to using only the Processing System (PS).

1. INTRODCUT

The escalating impacts of climate change and the global trend of industrialization have posed significant challenges to Taiwan's agricultural production. Coupled with the ongoing issues of diminishing and aging population in agricultural population, the situation has further exacerbated the already daunting challenges faced by agricultural production. To address these multifaceted challenges, Taiwan's agricultural sector has been continually striving for transformation, pushing towards technological progress and refinement. This transformation encompasses various aspects, ranging from the control of pests and diseases to the integration of technological innovations, all aimed at enhancing the yield and quality of agricultural products. In order to accurately comprehend the growth conditions of crops, manual field surveys used to be the essential approach in the past, although they also required a significant amount of manpower. With advancements in technology, aerial photograph has become one of the readily accessible sources of extensive data in agricultural environments. Through satellite, airplane, or Balloon, images of farmlands can be captured, enabling estimation of planted area, identification of crop types, and study of global vegetation changes under the influence of climate variations. The maturation of UAV technology has found substantial application in agriculture. The utilization of UAV for tasks such as pesticide spraying and field crop monitoring through remote sensing systems has significantly heightened agricultural production efficiency, minimized ecological impacts, and reduced labor requirements (Yang et al., 2021; Shih et al., 2021; Chen et al., 2023). With the rapid advancement of hardware and software, Artificial Intelligence (AI) has been widely applied in various fields. Among its applications, CNN are frequently employed in image classification, image recognition, and natural language processing tasks. However, its computation relies heavily on extensive matrix operations, which can lead to highly intricate computational processes. Thus, the selection of an appropriate computing platform becomes crucial. In comparison to Central Processing Unit (CPU) and Graphics Processing Unit (GPU), FPGA offer exceptional parallel processing capabilities, handling large volumes of data simultaneously. The FPGA's attributes of low power consumption, high flexibility, and short development cycles make it a preferred choice for accelerating neural network computations.

In this article, we utilize full-color images of rice paddies captured by UAV. Leveraging deep learning algorithms, we use semantic segmentation to compute the useful information, building models to locate and label rice paddies. Furthermore, a segment of the model is optimized using FPGA to reduce overall power consuming. According to the proposed approach, experimental results reveal that, when running on a desktop computer, the overall F_1-measure value of the model reached approximately 94%. When executed on the PYNQ-Z1 with the assistance of PL, the computation time decreased by 120 times compared to using only the PS for computation. In comparison to the desktop computer, the computation time was increase by approximately 88%, but the overall power consumption was only 1.66 watts on the PYNQ-Z1, albeit with a slight decrease in accuracy of 1%.

2. RESEARCH METHODOLOGY

2.1 Network Architecture

In this article, we refer to the Ref. (Shih et al., 2021) built the model using architectures such as ResNet (He, et al, 2016.) and atrous spatial pyramid pooling (ASPP) (Chen et al, 2018.). ResNet is used for downsampling, while ASPP is used to extract and combine features. The resulting features were subsequently upsampled to obtain the output. Then through the sigmoid function obtain the one-channel images.

2.2 Loss Functions

In the context of full-color images captured by UAV, the network architecture and loss function play a significant role in the detection and interpretation of transplant positions for rice paddies. In this article, we refer to the Ref. (Shih et al., 2021) and utilize a modified mean-square error loss function with the aim of achieving improved performance. The function can be defined as follows.

$$MLoss = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \,/\, \sum_{i=1}^{n}({y_i}^2 + {\hat{y}_i}^2) \tag{1}$$

where n, y, and $\hat{y}$ are the number of the data points, the ground truth value, and the predicted output value.

2.3 F_1-Measure

The F-measure (or F-score) is a common accuracy indicator used in machine learning and data analysis to assess model performance. Based on Table 1, the confusion matrix allows us to categorize the prediction results into four categories. Precision and recall can be calculated using Eq. (2) and Eq. (3) respectively, and by combining the two, we can obtain the F-measure using Eq. (4) (F_1-Measure, 2023).

Table 1. F_1-MEASURE

Predicted Class / Actual class	Positive (PP)	Negative (PN)
Positive (P)	True Positive (TP)	False Negative (FN)
Negative (N)	False Positive (FP)	True Negative (TN)

$$\text{Precision} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FP}) \tag{2}$$

$$\text{Recall} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN}) \tag{3}$$

$$F_1\text{-measure }\% = 2 \times (\text{Precision} \times \text{Recall})/(\text{Precision} + \text{Recall}) \times 100\% \tag{4}$$

2.4 Generate Register Transfer Level (RTL) code

Tensil is a toolset used for designing accelerators. It enables users to select the appropriate accelerator architectures based on the FPGA development board they are using. Tensil uses its RTL generator to produce RTL code, then use Xilinx Vivado for synthesis to create a Tensor Compute Unit (TCU) accelerator. The second branch is to compile the trained model to a Tensil binary consisting of TCU instructions. Finally, all the components are assembled and deployed on the development board for further testing and utilization. The advantage of Tensil lies in its ability to easily and effortlessly create an accelerator for your own model without the need for quantization or other forms of degradation (Tensil AI, 2022; Kim et al., 2023).

3. EXPERIMENTAL RESULTS

3.1 Dataset Description

The rice seeding dataset is from the Crop Location Auto-Labeling Competition (AI CUP 2021), AIdea Artificial Intelligence Collaboration Platform, Taiwan (The AIdea Artificial Intelligence Collaboration Platform, 2021). This dataset comprises 44 images(36 for training, 8 for testing) along with their corresponding CSV coordinate files. Depending on the varying durations of growth, it can be further categorized into two groups: those with a longer growth period (2000×3000 pixels) and those with a shorter growth period (1728×2304 pixels). As illustrated in Figure 1 and Figure 2.

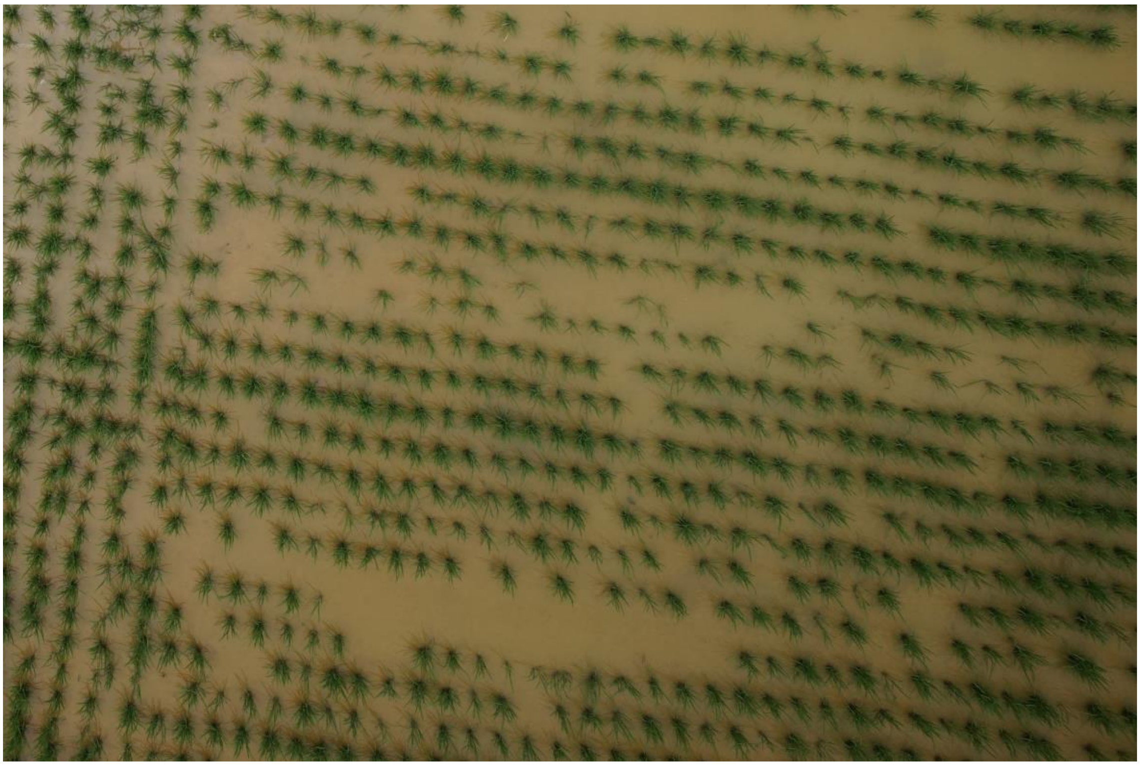


Figure 1. Rice seeding with a longer growth period.

Figure 2. Rice seeding with a shorter growth period.

3.2 Weight Conversion and Segmentation

Because the model compiler in Tensil supports weight file formats in onnx, whereas our trained model weights are in pth format, it is necessary to perform a conversion of the original weights before proceeding with compilation, we choose a part of Figure 1 to be the one predicting data as Figure 3. We also compared the converted weights to assess whether there were any significant differences in prediction results, as illustrated in Figure 4 and Figure 5. After weight conversion, there was minimal impact on the prediction results, indicating negligible loss. In this article, our emphasis is placed on accelerating the ResNet part of the model. During inference, we split the model and weights into two parts: the ResNet part and the remaining parts. When making predictions, we input the prediction image and store the feature maps computed through ResNet operations in the form of a txt file as tensors. Subsequently, we read the stored txt file for further computations and compare the final results, as illustrated in Figure 4 and Figure 6. The overall flow diagram is shown in Figure 7.

Figure 3. A part of Figure 1.

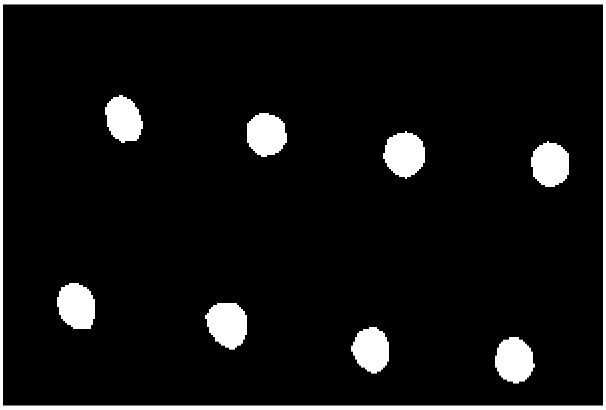


Figure 4. Prediction results using original weights.

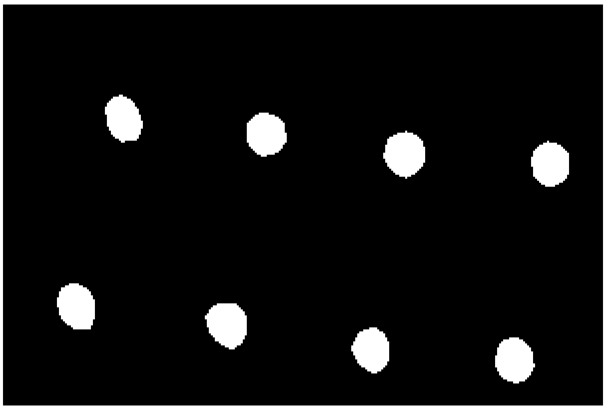


Figure 5. Prediction results using conversion weights.

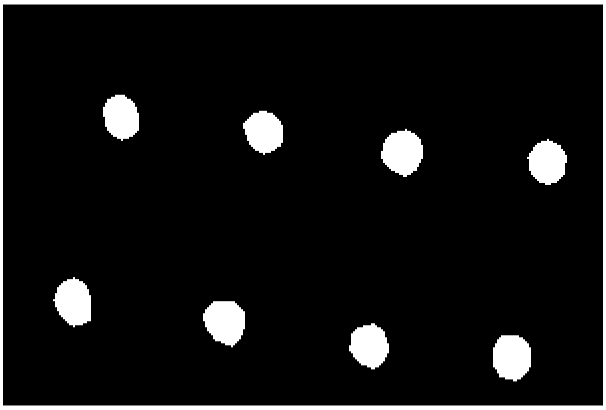


Figure 6. Prediction results using PYNQ-Z1.

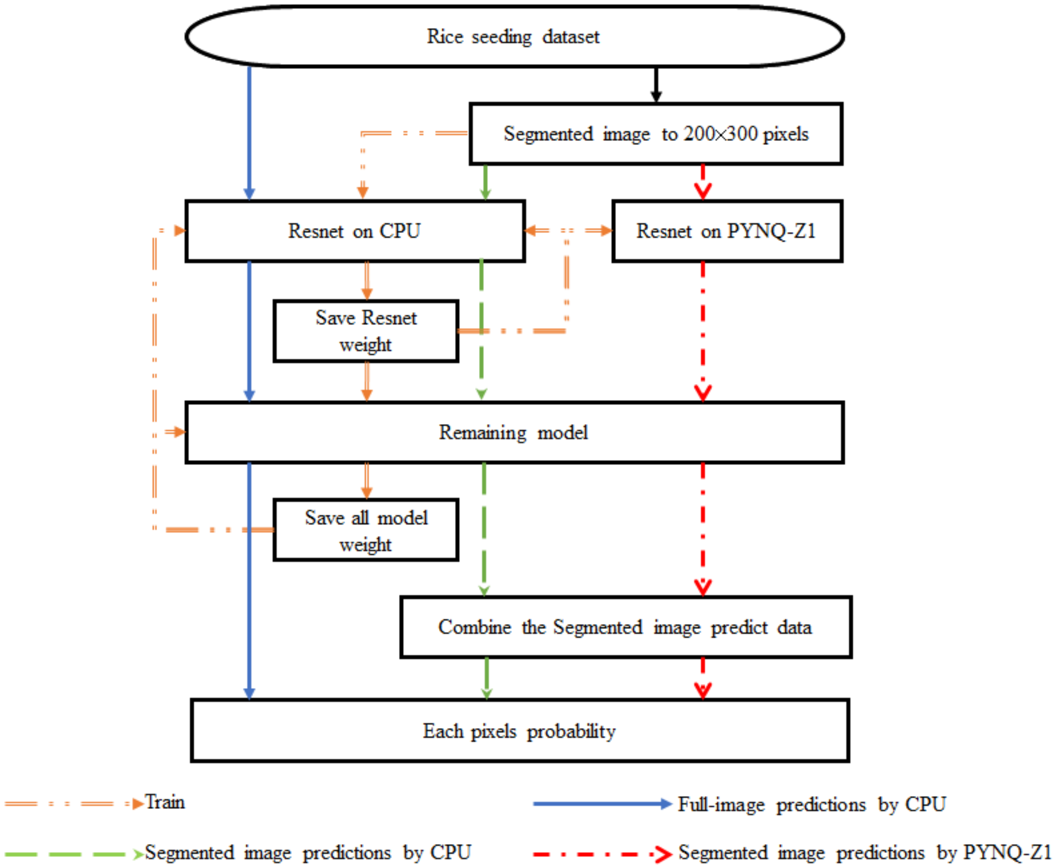


Figure 7. The flow diagram.

3.3 Predictive Image Segmentation

In this article, we utilized Xilinx's PYNQ-Z1 FPGA development board. This development board supports the Python language and enables command execution through the Jupyter Notebook, which facilitates the overall accelerator design process. However, due to hardware limitations, we need to divide the images (Figure 1, for example) into smaller segments of 200×300 pixels, as shown in Figure 3, for prediction. We then predict these segments individually and combine them to form a complete image, as illustrated in Figure 8, Figure 9, and Figure 10, these three images depict the results of different processing procedures and represent the most distinct portion of this image prediction. We assessed this impact by calculating the F_1-measure value for all three cases. Overall, segmented images resulted in a lower F_1-measure value compared to unsegmented images, with this difference decreasing to approximately 1-2% after applying Huffman circle and subsequent processing to all three cases, as illustrated in Figure 11. The white dot are the known points, the blue dots are after the Huffman circle processing in Figure 8, the green dots are after the Huffman circle processing in Figure 9, and the red dots are after the Huffman circle processing in Figure 10.

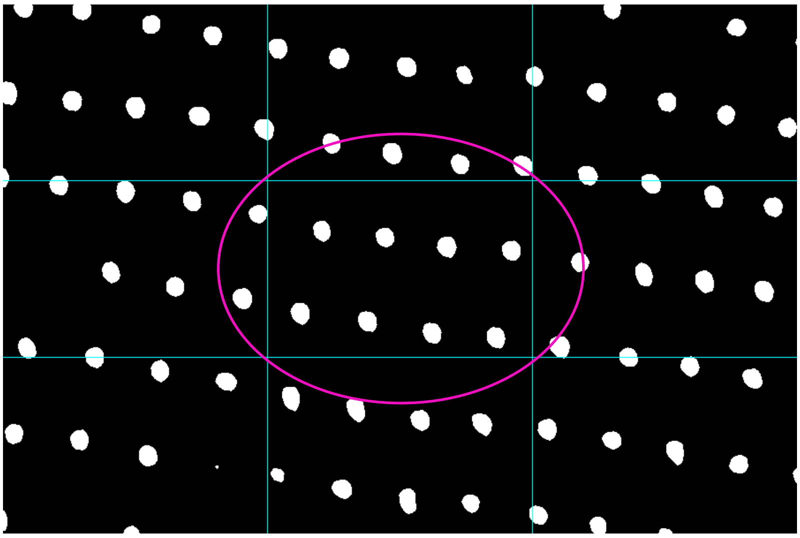


Figure 8. Full-image predictions by CPU.
(The circled area is Figure 3 position.)

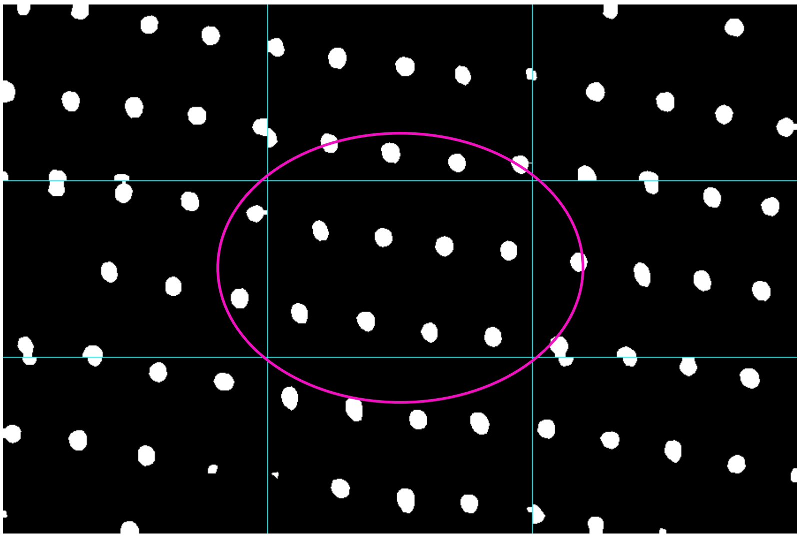


Figure 9. Segmented image predictions and combine to full-image by CPU.
(The circled area is Figure 3 position.)

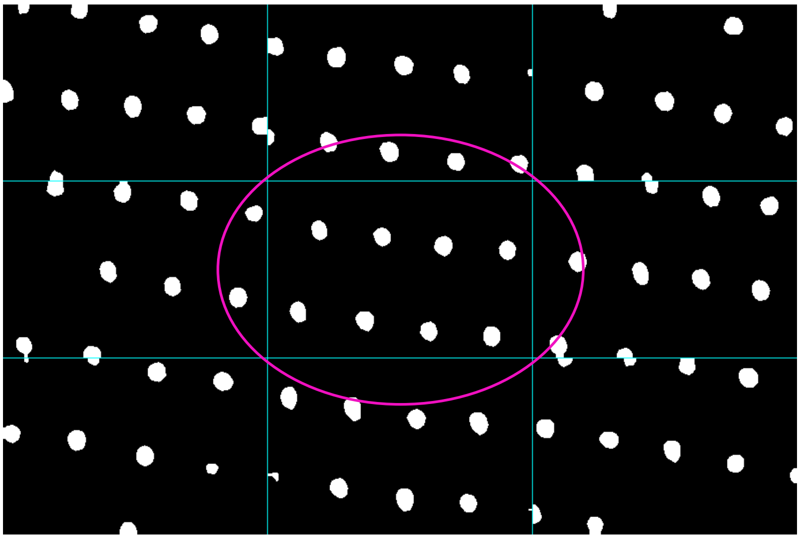


Figure 10. Segmented image predictions and combine to full-image by PYNQ-Z1.
(The circled area is Figure 3 position.)

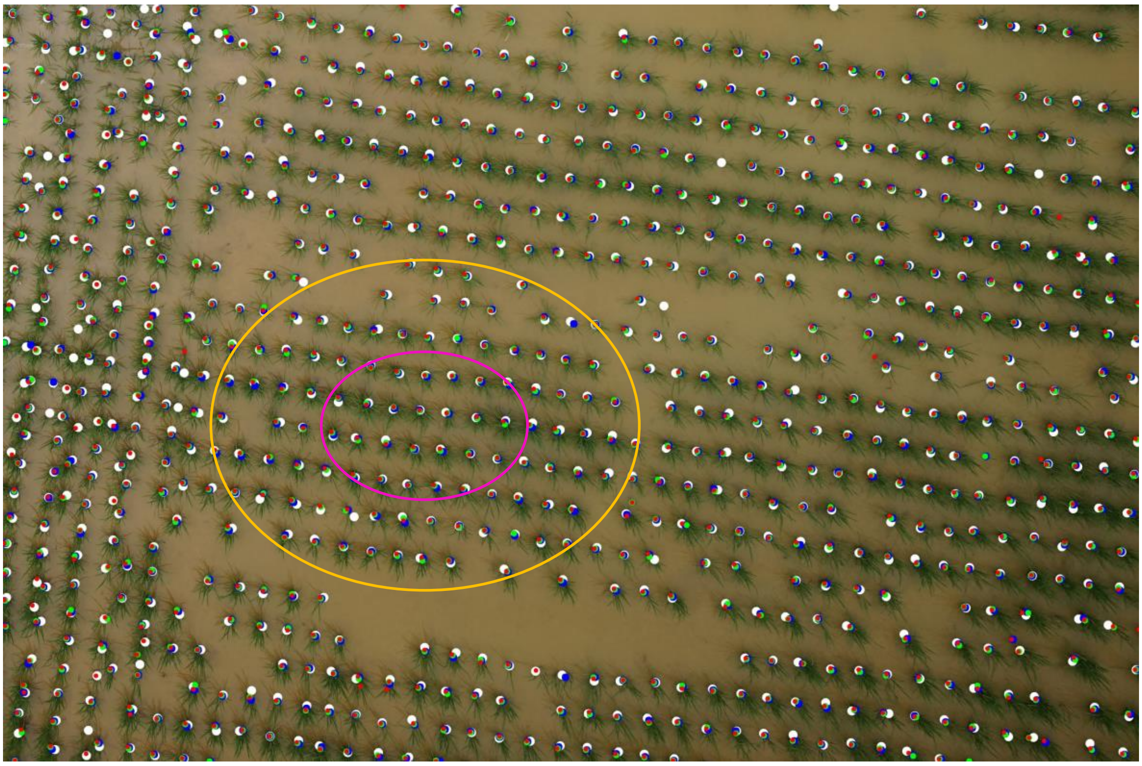


Figure 11. Three kinds of predicted results and the known areas.
(The pink circled area is position of Figure 3. The yellow circled area is position of Figure 8, Figure 9, Figure 10.)

3.4 Performance Evaluation

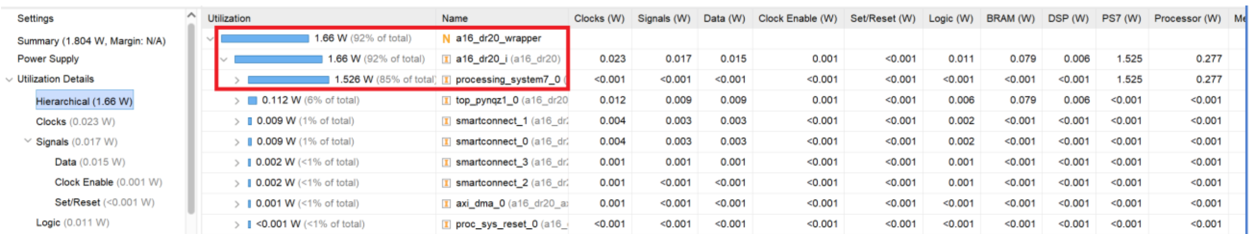
According to the proposed approach, experimental results indicate that the model optimized by FPGA significantly outperforms the solely used PS on PYNQ-Z1 in terms of computational speed, although it is slightly slower than a CPU, as illustrated in Table 2. In terms of accuracy, utilizing FPGA for segmented image predictions results in a marginal decrease of 1% compared to using a CPU for full-image predictions, as illustrated in Table 3. However, it consumes only 1.66 watts of power as illustrated in Figure 12.

Table 2. Computational speed

	Ryzen 5950x	PS on PYNQ-Z1	PS+PL on PYNQ-Z1
Time spent per image (200 × 300 pixels)	0.267 s	60.657s	0.502 s

Table 3. Accuracy

	Full images prediction by CPU	Segmented image predictions and combine to full-image by CPU	Segmented image predictions and combine to full-image by PYNQ-Z1
F_1-measure value	93.83 %	92.47 %	93.06 %



Utilization	Name	Clocks (W)	Signals (W)	Data (W)	Clock Enable (W)	Set/Reset (W)	Logic (W)	BRAM (W)	DSP (W)	PS7 (W)	Processor (W)
1.66 W (92% of total)	a16_dr20_wrapper										
1.66 W (92% of total)	a16_dr20_i (a16_dr20)	0.023	0.017	0.015	0.001	<0.001	0.011	0.079	0.006	1.525	0.277
1.526 W (85% of total)	processing_system7_0	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	1.525	0.277
0.112 W (6% of total)	top_pynqz1_0 (a16_dr20	0.012	0.009	0.009	0.001	<0.001	0.006	0.079	0.006	<0.001	<0.001
0.009 W (1% of total)	smartconnect_1 (a16_dr	0.004	0.003	0.003	<0.001	<0.001	0.002	<0.001	<0.001	<0.001	<0.001
0.009 W (1% of total)	smartconnect_0 (a16_dr	0.004	0.003	0.003	<0.001	<0.001	0.002	<0.001	<0.001	<0.001	<0.001
0.002 W (<1% of total)	smartconnect_3 (a16_dr	0.001	0.001	0.001	<0.001	<0.001	0.001	<0.001	<0.001	<0.001	<0.001
0.002 W (<1% of total)	smartconnect_2 (a16_dr	0.001	<0.001	<0.001	<0.001	<0.001	0.001	<0.001	<0.001	<0.001	<0.001
0.001 W (<1% of total)	axi_dma_0 (a16_dr20_a	0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001
<0.001 W (<1% of total)	proc_sys_reset_0 (a16_	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001

Figure 12. Power consumption on PYNQ-Z1.

4. CONCLUSION

Based on the experimental results, we observe that our completed neural model, running on PYNQ-Z1, while not as fast as the 'Ryzen 5950x' and exhibiting a slight decrease in accuracy, still performs well in terms of power consumption, consuming only 1.66 Watts. This demonstrates the superiority of PYNQ-Z1 for low-power applications, especially in resource-constrained environments such as drones. However, we find that by implementing customized neural models on FPGAs, we achieve computational performance beyond the PS side of PYNQ-Z1. Experiments show that FPGAs still excel in terms of computation speed, with speedups of up to 120 times, which is crucial for real-time, demanding applications. At the same time, our FPGA implementation maintains extremely low power consumption levels, making it a highly competitive option. The success of this study demonstrates the great potential of FPGA technology for image processing and neural network applications. Our results provide a valuable reference for future research, especially in drone applications as well as agricultural automation.

ACKNOWLEDGMENT

The rice seedling datasets are supported by the Crop Location Auto-Labeling Competition (AI CUP 2021), AIdea Artificial Intelligence Collaboration Platform, Taiwan.
This work was supported in part by the Ministry of Science and Technology, Taiwan (R.O.C.), under Grant numbers of MOST 110-2121-M-033-001 and MOST 111-2121-M-033-001, and in part by the National Science and Technology Council, Taiwan (R.O.C.), under Grant numbers of NSTC 112-2121-M-033-001 and NSTC 112-2813-C-033-080-E.

REFERENCES

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A., 2018. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4), pp. 834-848.

Chen, Y.-C., Chiou, Y.-S., Shih, M.-J., 2023. Interpretation of transplanted positions based on image super-resolution approaches for rice paddies. In *Proc. 2023 Sixth International Symposium on Computer, Consumer and Control (IS3C)*, pp. 358-361.

F_1-Measure, 2023. (Available: https://en.wikipedia.org/wiki/F-score, Accessed: September 2023)

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In *Proc. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770-778.

Kim, V. H., Choi, K. K., 2023. A reconfigurable CNN-based accelerator design for fast and energy-efficient object detection system on mobile FPGA. *IEEE Access*, 11, pp. 59438-59445.

Shih, M.-J., Chiou, Y.-S., Chen, Y.-C., 2021. Detection and interpretation of transplanted positions using drone's eye-view images for rice paddies. In *Proc. 2021 IEEE 3rd Eurasia Conference on IOT, Communication and Engineering (ECICE)*, pp1-5.

Tensil AI, 2022. (Available: https://www.tensil.ai, Accessed: September 2023)

The AIdea Artificial Intelligence Collaboration Platform, 2021 (Available: https://aidea-web.tw/about, Accessed: September 2023)

Yang, M.-D., Tseng, H.-H., Hsu, Y.-C., Yang, C.-Y., Lai, M.-H., Wu, D.-H., 2021. A UAV open dataset of rice paddies for deep learning practice. *Remote Sensing*, 13(7), pp1358-1374.